

A Clearer Picture of Total Variation Blind Deconvolution

Daniele Perrone, Paolo Favaro, *Member, IEEE*

Abstract—Blind deconvolution is the problem of recovering a sharp image and a blur kernel from a noisy blurry image. Recently, there has been a significant effort on understanding the basic mechanisms to solve blind deconvolution. While this effort resulted in the deployment of effective algorithms, the theoretical findings generated contrasting views on why these approaches worked. On the one hand, one could observe experimentally that alternating energy minimization algorithms converge to the desired solution. On the other hand, it has been shown that such alternating minimization algorithms should fail to converge and one should instead use a so-called Variational Bayes approach. To clarify this conundrum, recent work showed that a good image and blur prior is instead what makes a blind deconvolution algorithm work. Unfortunately, this analysis did not apply to algorithms based on total variation regularization. In this manuscript, we provide both analysis and experiments to get a clearer picture of blind deconvolution. Our analysis reveals the very reason why an algorithm based on total variation works. We also introduce an implementation of this algorithm and show that, in spite of its extreme simplicity, it is very robust and achieves a performance comparable to the top performing algorithms.

Index Terms—Deblurring, blind deconvolution, total variation.



Blind deconvolution is the problem of recovering a signal and a degradation kernel from their noisy convolution. This problem is found in diverse fields such as astronomical imaging, medical imaging, (audio) signal processing, and image processing. More recently, blind deconvolution has received renewed attention due to the emerging need for removing motion blur in images captured by mobile phones [1]. Yet, despite over three decades of research in the field (see [2] and references therein), the design and analysis of a principled, stable and robust algorithm that can handle real images remains a challenge. However, present-day progress has shown that recent models for sharp images and blur kernels can yield remarkable results [1], [3], [4], [5], [6].

Many of these recent approaches are evolutions of the variational formulation [7]. A common element in these methods is the explicit use of priors for both blur and sharp image to encourage smoothness in the solution. Among these recent methods, *total variation* emerged as one of the most popular priors [8], [9]. Such popularity is probably due to its ability to encourage gradient sparsity, a property that can describe many signals of interest well [10].

However, recent work by Levin et al. [11] has shown that the joint optimization of both image and blur kernel can have the no-blur solution¹ as its global minimum. That is to say, blind deconvolution algorithms that use a total variation prior either are local minimizers and, hence, require a lucky initial guess, or they cannot depart too much from the initial blurry image. Nonetheless, algorithms based on the joint optimization

of blur and sharp image show good convergence behavior even when initialized with the no-blur solution [8], [3].

This incongruence called for an in-depth analysis of total variation blind deconvolution (TVBD). As we show in the next sections, the answer is not as straightforward as one might have hoped. Firstly, we confirm both experimentally and analytically the analysis of Levin et al. [11]. Secondly, we also find that the algorithm of Chan and Wong [8] converges to the desired solution, even when starting at the no-blur solution. The answer to this puzzle lies in the specific implementation of [8], as it does not minimize the originally-defined energy. This algorithm, as commonly done in many other algorithms, separates some constraints from the gradient descent step and then applies them sequentially. When the cost functional is convex this alteration may not have a major impact. However, in blind deconvolution, where the cost functional is not convex, this completely changes the convergence behavior. Indeed, we show that if one imposed all the constraints simultaneously then the algorithm would never leave the no-blur solution independently of the regularization amount.

To further demonstrate our findings, we implement a BD algorithm without the use of all recent improvements, such as filtering [3], [4], [11], blur kernel prior [8], [9] or edge enhancement [4], [5], and show that applying sequentially the constraints on the blur kernel is sufficient to avoid the no-blur solution. We also show that the use of a filtered version of the original signal may be undesirable and that the use of exact boundary conditions can improve the results. Finally, we apply the algorithm on currently available datasets, compare it to the state of the art methods and show that, notwithstanding its simplicity, it achieves a comparable performance to the top performers.

• D. Perrone and P. Favaro are with the Institute of Computer Science and Applied Mathematics, University of Bern, Nebrückstrasse 10, 3012 Bern, Switzerland.
E-mail: perrone@iam.unibe.ch; paolo.favaro@iam.unibe.ch

1. The so-called no-blur, or trivial, solution in the format (*sharp image*, *blur kernel*) is the pair (*blurry image*, *Dirac delta*).

1 BLUR MODEL AND PRIORS

Suppose that a blurry image f can be modeled by

$$f = k_0 * u_0 + n \quad (1)$$

where k_0 is a blur kernel, u_0 a sharp image, n noise and $k_0 * u_0$ denotes convolution between k_0 and u_0 . Given only the blurry image, one might want to recover both the sharp image and the blur kernel. This task is called *blind deconvolution*. A classic approach to this problem is to maximize the posterior distribution (MAP _{u,k})

$$\arg \max_{u,k} p(u, k | f) = \arg \max_{u,k} p(f | u, k) p(u) p(k). \quad (2)$$

where $p(f | u, k)$ models the noise affecting the blurry image, $p(u)$ models the distribution of typical sharp images, and $p(k)$ is the prior knowledge about the blur function. Typical choices for $p(f | u, k)$ are the Gaussian distribution [1], [6] or the exponential distribution [5]. In the following discussion we will assume that $p(f | u, k)$ is modeled by a Gaussian distribution.

Through some standard transformations and assumptions, problem (2) can be written also as a regularized minimization

$$\arg \min_{u,k} \|k * u - f\|_2^2 + \lambda J(u) + \gamma G(k) \quad (3)$$

where the first term corresponds to $p(f | u, k)$, the functionals $J(u)$ and $G(k)$ are the smoothness priors for u and k (for example, Tikhonov regularizers [12] on the gradients), and λ and γ two nonnegative parameters that weigh their contribution. Furthermore, additional constraints on k , such as positivity of its entries and integration to 1, can be included. For any $\lambda > 0$ and $\gamma > 0$ the cost functional will not have as global solution neither the true solution (k_0, u_0) nor the *no-blur solution* ($k = \delta, u = f$), where δ denotes the Dirac delta. Indeed, eq. (3) will find an optimal tradeoff between the data fitting term and the regularization term. Nonetheless, one important aspect that we will discuss later on is that both the true solution (k_0, u_0) and the *no-blur solution* make the data fitting term in eq. (3) equal to zero. Hence, we can compare their cost in the functional simply by evaluating the regularization terms. Notice also that the minimization objective in eq. (3) is non-convex, and, as shown in Fig. 3, has several local minima.

2 PRIOR WORK

To solve problem (3) one has to carefully choose the functions $J(u)$ and $G(k)$. A common choice is the ℓ_p norm of u and k or of some filtered versions of u and k .

$G(k)$ has been defined as the ℓ_2 norm of k [7], [5], [4], a sparsity-inducing norm ($p \leq 1$) [1], [3] or a constant [6]. Nonetheless, its contribution to the estimation of the desired solution is so far marginal. In fact, some methods successfully solve problem (3) by setting $G(k) = \text{const}$. Yet, it has been shown that its use may help avoid undesired local minima [13].

The regularization term for the sharp image $J(u)$ instead has a stronger impact on the performance of the algorithm, since it helps choose a sharp image over a blurry one. You and Kaveh [7] have proposed to use the ℓ_2 norm of the derivatives of u . Unfortunately, this norm is not able to model the sparse

nature of common image gradients and results in sharp images that are either oversmoothed or have ringing artifacts. Yet, the ℓ_2 norm has the desirable property of being efficient to minimize. Cho and Lee [4] and Xu and Jia [5] have reduced the generation of artifacts while still retaining its computational efficiency by using heuristics to select sharp edges.

An alternative to the ℓ_2 norm is the use of *total variation* (TV) [8], [9], [14], [15]. TV regularization was firstly introduced for image denoising in the seminal work of Rudin, Osher and Fatemi [16], and since then it has been applied successfully in many image processing applications. Total variation is typically defined via two different formulations. Its anisotropic version is the sum of the ℓ_1 norms of the components of the gradient of u , while its isotropic version is the ℓ_2 norm of the gradient of u .

You and Kaveh [9] and Chan and Wong [8] have proposed the use of TV regularization in blind deconvolution on both u and k . They also consider the following additional constraints to enhance the convergence of their algorithms

$$\|k\|_1 \doteq \sum_{\mathbf{x}} |k(\mathbf{x})| = 1, \quad k(\mathbf{x}) \geq 0, \quad u(\mathbf{x}) \geq 0 \quad (4)$$

where with \mathbf{x} we denote either 1D or 2D coordinates. He *et al.* [15] have incorporated the above constraints in a variational model, claiming that this enhances the stability of the algorithm. A different approach is a strategy proposed by Wang *et al.* [17] that seeks for the desired local minimum by using downsampled reconstructed images as priors during the optimization in a multi-scale framework.

TV regularization has been widely popularized because it models natural image gradients well [10]. Wipf and Zhang [18] have recently argued that $J(u)$ should not merely try to model statistics of sharp images, but, rather, it should have a strong discriminative power in separating sharp from blurry images. This is ideally achieved by using the ℓ_0 pseudo-norm on the image gradients. Unfortunately its exact minimization requires solving an NP-hard problem. To make the problem tractable some methods have proposed to use approximations of the ℓ_0 norm [19], [20]. Yet, the proposed approximations are non-convex functions that require careful minimization strategies to avoid local minima.

The algorithms presented so far are all successful implementation of the MAP _{u,k} formulation in (2), nonetheless Levin *et al.* [11] have shown that using an ℓ_p norm of the image gradients with any $p > 0$ and a uniform distribution for the blur, the MAP _{u,k} approach favors the no-blur solution ($u = f, k = \delta$), for images blurred with a large enough blur. They also argue that the success of existing MAP _{u,k} methods is due to various heuristics or reweighing strategies employed during the optimization of (3).

Because of the above concerns, Levin *et al.* [11] look at a different strategy that marginalizes over all possible sharp images u . They solve the following MAP _{k} problem

$$\arg \max_k p(k | f) = \arg \max_k \int p(f | u, k) p(u) p(k) du, \quad (5)$$

where the sharp image u is estimated by solving a convex problem and where k is given from the previous step. They

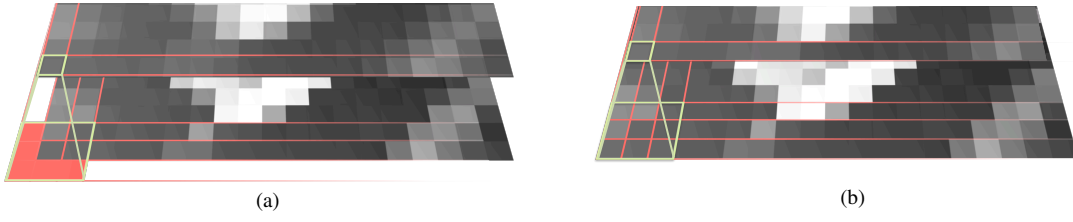


Fig. 1. Illustration of the convolution operators (8) and (9) assuming a blur support of 3×3 . a) With the use of (8) we assume that the blurry image f and the sharp image u have the same support, therefore we must choose how the values at the boundaries of u are defined (red pixels); b) With (9) we assume that f has a smaller support than u , therefore the pixels of f are completely defined by the pixels of u .

have shown that, for sufficiently large images, the MAP_k approach converges to the true solution. Since the right hand side of problem (5) is difficult to compute, it is common to use a Variational Bayesian approach (VB) where one aims at finding an approximation of the probability $p(k|f)$ [21], [1], [22], [18], [6].

In recent work, Wipf and Zhang [18] have shed new light on the $\text{MAP}_{u,k}$ vs MAP_k dispute. They have shown that the VB formulation commonly used to solve the MAP_k is equivalent to a non-conventional $\text{MAP}_{u,k}$ approach, and that ℓ_p priors with $p \leq 0.5$ are able to favor sharp images. They also argue that a VB approach is still preferable because it is more able to avoid local minima compared to a classical $\text{MAP}_{u,k}$ approach.

The work of Wipf and Zhang [18] has focused on ℓ_p priors with $p < 1$ and given novel insights on the mechanism that make them work. Our work complements the results of Wipf and Zhang [18] and focuses on the total variation ($p = 1$) prior. We confirm the theoretical results of Levin *et al.* [11] and show that an apparently harmless delayed normalization induces a scaling of the signal that ultimately results in the success of total variation based algorithms. This shows that filtered version of the images [3], [4], [11], blur kernel prior [8], [9], edge enhancement [4], [5] or any other additional strategy are not necessary for solving blind deconvolution.

3 PROBLEM FORMULATION

Similarly to many methods in the literature in this paper we consider the problem (3) in a discrete setting. By considering the constraints in eq. (4) and by setting $\gamma = 0$, we study the following minimization

$$\begin{aligned} \min_{u,k} \quad & \|k * u - f\|_2^2 + \lambda J(u) \\ \text{subject to} \quad & k \succcurlyeq 0, \quad \|k\|_1 = 1 \end{aligned} \quad (6)$$

where $J(u) = \|\nabla u\|_{2,1}$ or $J(u) = \|\nabla u\|_{1,1}$, $\nabla u \doteq [u_x \ u_y]^T$ is the discrete gradient of u , $\mathbf{x} \doteq [x \ y]^T$, and²

$$\|\nabla u\|_{p,q} \doteq (\sum_{\mathbf{x}} \|\nabla u(\mathbf{x})\|_p^q)^{\frac{1}{q}}. \quad (7)$$

To keep the analysis simple we have stripped the formulation of all unnecessary improvements such as using a basis of filters in $J(u)$ [3], [4], [11], or performing some selection of regions by reweighing the data fitting term [5], or enhancing the edges

of the blurry image f [5], [4]. Compared to previous methods, we do not use any regularization on the blur kernel ($\gamma = 0$).

The formulation in eq. (6) involves the minimization of a constrained non-convex optimization problem. Also, notice that if (u, k) is a solution, then $(u(\mathbf{x} + \mathbf{c}), k(\mathbf{x} + \mathbf{c}))$ are solutions as well for any $\mathbf{c} \in \mathbb{R}^2$. If the additional constraints on k were not used, then the ambiguities would also include $(\alpha_1 u, \frac{1}{\alpha_1} k)$ for non zero α_1 .

3.1 Convolution Models and Notation

The convolution operator in the minimization (6) usually requires some assumptions on the boundaries of the image. We instead propose a formulation that does not make any boundary assumptions.

Let u and f be matrices with the same support $m \times n$, and k a matrix with support $h \times w$. Typically the discrete convolution of u and k is defined by

$$f = (u * k)[i, j] \doteq \sum_{r=-\lfloor h/2 \rfloor}^{\lceil h/2 \rceil - 1} \sum_{c=-\lfloor w/2 \rfloor}^{\lceil w/2 \rceil - 1} u[i-r, j-c] k[r, c] \quad (8)$$

for $i = 1, \dots, m$, $j = 1, \dots, n$, where some assumptions are made on the values outside the support of u (Fig. 1a). Commonly used assumptions in the literature are: *symmetric*, where the boundary of the image is mirrored to fill the additional frame around the image; *periodic*, where the image is padded with a periodic repetition of the boundary; *replicate*, where the borders continue with a constant value. The periodic assumption is particularly convenient because it allows the use of the circular convolution theorem and the Discrete Fourier Transform (DFT) to achieve a fast performance. Because real images are rarely periodic, Liu and Jia [23] have proposed to extend the size of the blurry image to make it periodic. Nonetheless, each of the above choices is an approximation of the real physical phenomenon and therefore it introduces an error in the reconstruction of the sharp image.

In this paper we propose to use a different approach, where the blurry image f has support $m - h + 1 \times n - w - 1$. In this case we consider a different convolution operator, denoted by \circ , as³

$$f = (u \circ k)[i, j] \doteq \sum_{r=0}^{h-1} \sum_{c=0}^{w-1} u[i+r, j+c] k_{-}[r, c] \quad (9)$$

for $i = 1, \dots, m - h + 1$, $j = 1, \dots, n - w + 1$ and where $k_{-}[r, c] = k[h-r, w-c]$ (Fig. 1b). Notice that $k \circ u \neq u \circ k$

2. $\|\cdot\|_p$ is the usual vector p -norm defined as $\|v\|_p \doteq (\sum_n v_n^p)^{1/p}$.

3. Notice that this formulation is MATLAB's **valid** convolution.

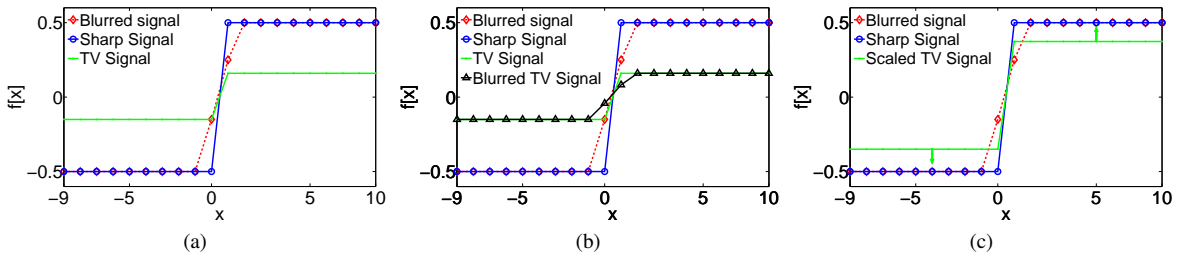


Fig. 2. Illustration of proposition 4.5 and proposition 4.4 (best viewed in color). The original step function is denoted by a solid-blue line. The TV signal (green-solid) is obtained by solving $\arg \min_u \|u - f\|_2^2 + \lambda J(u)$. In (a) we show how the TV denoising algorithm reduces the contrast of a blurred step function (red-dotted). In (b) we illustrate proposition 4.4: If the constraints on the blur are enforced, any blur different from the Dirac delta increases the distance between the input blurry signal and the blurry TV signal (black-solid). In (c) we illustrate proposition 4.5: In the second step of the PAM algorithm, estimating a blur kernel without a normalization constraint is equivalent to scaling the TV signal.

in general. Also, $u \circ k$ is not defined if the support of k is too large ($h > m + 1$ and $w > n + 1$).

In the following we will choose $J(u)$ to be the isotropic total variation $J(u) = \|\nabla u\|_{2,1}$. By incorporating the above considerations in a discrete setting, the problem in (6) can be written as

$$\min_{u,k} \sum_{\mathbf{x} \in \mathcal{F}} ((k \circ u)[\mathbf{x}] - f[\mathbf{x}])^2 + \sum_{\mathbf{x} \in \mathcal{U}} \|\nabla u[\mathbf{x}]\|_2 \quad (10)$$

subject to $k \succeq 0$, $\|k\|_1 = 1$

where $\mathcal{F} = \{1, \dots, m + h - 1\} \times \{1, \dots, n + w - 1\}$ and $\mathcal{U} = \{1, \dots, m\} \times \{1, \dots, n\}$ ⁴.

Since in (10) the domain of the sharp image u is larger than the domain of the blurry image f , solving (10) requires the estimation of more variables than measurements. This problem is tackled by the total variation term that imposes a weak smoothness constraint beyond the boundary of f , instead of the hard assumptions needed when using eq. (8). We will show that using this formulation gives better results compared to the typical approach that uses eq. (8).

While the analysis in our previous paper [24] is based on the circular convolution operator, this paper is entirely based on the convolution operator defined in eq. (9).

4 ANALYSIS OF TOTAL VARIATION BLIND DECONVOLUTION

Recent analysis [11], [18] has shown that the total variation prior (and some variations on the same theme) is not suitable for blind deconvolution as it favors blurry solutions. Nonetheless, many iterative algorithms successfully employ this prior. This section provides analysis to understand why these total variation-based blind deconvolution methods (and in particular [8]) work. Here we give an overview of the contents.

Limitations of TV as a prior for blind deconvolution. Our first step is to confirm and extend an earlier result of Levin et al [11]. We demonstrate that a family of priors (which include total variation) favors blurry over sharp images (Section 4.1). This means that they should not be used as regularizers in

an energy minimization formulation of blind deconvolution. Despite this result, the TVBD algorithm of Chan and Wong [8] uses such prior and works well. Because the use of iterative methods, alternating minimization and total variation priors is common practice in the literature, we devote a large part of the analysis to understanding the interplay of these components.

Analysis and experiments via a 1D toy example. We formulate an Alternating Minimization (AM) algorithm that precisely implements a gradient descent on the TVBD energy (Section 4.2). Then, we show that this formulation can never converge to the sharp solution if we initialize the algorithm with the no-blur solution. This reinforces the idea that total variation is not a suitable prior for blind deconvolution. We do so in two ways: 1) analytically, by obtaining explicit analytical solutions in 1D with a step signal as sharp input (Sections 4.2.1 and 4.2.2) and 2) experimentally, by showing the energies (and minima) of the whole solution space of the blur (Section 4.3.1).

An unconventional solution to the TV prior limitations.

We introduce a bare-bone version of TVBD [8] (we remove the smoothness prior on the blur) and call it Projected Alternating Minimization (PAM) algorithm, because the constraints on the blur kernel are imposed separately in a delayed step (Section 4.3.1). We show in Fig. 3 and with propositions 4.4 and 4.5 that PAM does not minimize the original TVBD energy. Thus, PAM is not a valid gradient descent method. Notwithstanding, this algorithm can find the correct solution even when initialized with the no-blur solution. We also show that there exists a family of 1D step signals which cannot be reconstructed once blurred (Section 4.4). However, because general 1D signals are made of several (different) steps, the likelihood that all the steps lie in this family is close to nil.

The role of regularization. Finally, in Section 4.5 we highlight the role of the regularization parameter λ and further stress the importance of the scaling principle to make the PAM algorithm succeed.

4.1 Limitations of Total Variation as a Prior for Blind Deconvolution

In a recent work, Levin *et al.* [6] have shown that total variation favors the no-blur solution over the true solution. The

4. Notice that \mathcal{F} represents the support of f and \mathcal{U} the support of u .

proof is based on observing that when there is no noise ($n = 0$) then the energy in eq. (6) $\|k * u - f\|_2^2 + \lambda J(u) = \lambda J(u)$ both with the no-blur solution $u = f, k = \delta$ and the true solution $u = u_0, k = k_0$. Thus, to determine which of the two solutions is preferred by the minimization procedure, they just compare the value of the regularization term $J(u)$ in the two cases. They showed that when $J(u) = \sum_{\mathbf{x}} |u_x(\mathbf{x})|^p + |u_y(\mathbf{x})|^p$, for any $p > 0$ and either the true blur k_0 has a large support or $\|k_0\|_2^2 \ll 1$, then $J(f) \leq J(u_0)$ so that the preferred solution is (f, δ) .

In the following proposition we show that the above result is also true for any kind of blur kernel. In our analysis, we consider the ℓ_p norm of the image derivatives with $p \geq 1$ as image prior.

Proposition 4.1: Let $J(u) = \|\nabla u\|_{p,p}$, with $p \in [1, \infty]$, f be the noise-free input blurry image ($n = 0$) and u_0 the sharp image. Then, $J(f) \leq J(u_0)$.

Proof: Because f is noise-free, $f = k_0 * u_0$; since the convolution and the gradient are linear operators, we have

$$J(f) = \|\nabla(k_0 * u_0)\|_{p,p} = \|k_0 * \nabla u_0\|_{p,p} \quad (11)$$

where

$$\|k_0 * \nabla u_0\|_{p,p} = \left(\sum_{\mathbf{x}} \left| \sum_{\mathbf{y}} k_0(\mathbf{y} - \mathbf{x}) u_x(\mathbf{x}) \right|^p + \left| \sum_{\mathbf{y}} k_0(\mathbf{y} - \mathbf{x}) u_y(\mathbf{x}) \right|^p \right)^{\frac{1}{p}} \quad (12)$$

by Jensen's inequality we have

$$\begin{aligned} \sum_{\mathbf{x}} \left| \sum_{\mathbf{y}} k_0(\mathbf{y}) u_x(\mathbf{x} - \mathbf{y}) \right|^p &\leq \sum_{\mathbf{x}, \mathbf{y}} k_0(\mathbf{y}) |u_x(\mathbf{x} - \mathbf{y})|^p \leq \\ \sum_{\mathbf{y}} k_0(\mathbf{y}) \sum_{\mathbf{x}} |u_x(\mathbf{x} - \mathbf{y})|^p &= \sum_{\mathbf{y}} k_0(\mathbf{y}) \sum_{\mathbf{x}} |u_x(\mathbf{x})|^p = \\ &= \sum_{\mathbf{x}} |u_x(\mathbf{x})|^p \end{aligned} \quad (13)$$

and similarly for u_y we have $\sum_{\mathbf{x}} \left| \sum_{\mathbf{y}} k_0(\mathbf{y}) u_y(\mathbf{x} - \mathbf{y}) \right|^p \leq \sum_{\mathbf{x}} |u_y(\mathbf{x})|^p$. From these inequalities it follows that

$$J(f) = \|k_0 * \nabla u_0\|_{p,p} \leq \|\nabla u_0\|_{p,p} \doteq J(u_0). \quad (14)$$

□

Since the first term (the data fitting term) in problem (6) is zero for both the no-blur solution (f, δ) and the true solution (u_0, k_0) , proposition 4.1 states that the no-blur solution has always a smaller, or at most equivalent, cost than the true solution. Notice that proposition 4.1 is also valid for any $J(u) = \|\nabla u\|_{p,p}^r$ for any $r > 0$. Thus, it includes as special cases the Gaussian prior $J(u) = \|\nabla u\|_{2,2}^2$, when $p = 2, r = 2$, and the anisotropic total variation prior $J(u) = \|u_x\|_1 + \|u_y\|_1$, when $p = 1, r = 1$.

Proposition 4.1 highlights a strong limitation of the formulation (6): *The exact solution can not be retrieved when an iterative minimizer is initialized at the no-blur solution.*

4.2 The Alternating Minimization (AM) Algorithm

To better understand the behavior of a total variation-based blind deconvolution algorithm we consider an *alternating minimization* algorithm that minimizes (10). The solution is found by alternating between the estimation of the sharp image

given the kernel and the estimation of the kernel given the sharp image. This approach, which we call the AM algorithm, requires solving an unconstrained convex problem in u

$$u^{t+1} \leftarrow \arg \min_u \sum_{\mathbf{x} \in \mathcal{F}} ((k^t \circ u)[\mathbf{x}] - f[\mathbf{x}])^2 + \sum_{\mathbf{x} \in \mathcal{U}} \|\nabla u[\mathbf{x}]\|_2 \quad (15)$$

and a constrained convex problem in k

$$\begin{aligned} k^{t+1} \leftarrow & \arg \min_k \sum_{\mathbf{x} \in \mathcal{F}} ((k \circ u^{t+1})[\mathbf{x}] - f[\mathbf{x}])^2 \\ & \text{subject to } k \succcurlyeq 0, \quad \|k\|_1 = 1. \end{aligned} \quad (16)$$

An explicit form of the solution of the AM algorithm is elusive, but when the algorithm is initialized at the no-blur solution, the first step of the AM algorithm requires solving a denoising problem

$$\hat{u} \leftarrow \arg \min_u \sum_{\mathbf{x} \in \mathcal{F}} (u[\mathbf{x}] - f[\mathbf{x}])^2 + \sum_{\mathbf{x} \in \mathcal{U}} \|\nabla u[\mathbf{x}]\|_2. \quad (17)$$

The total variation denoising algorithm (17) has been widely studied in the literature, and its analysis can give key insights on the behavior of the AM algorithm. In the next section we study this problem and present an important building block for the other results presented in the paper.

4.2.1 Analysis of 1D Total Variation Denoising

In this section we look at the solution of a 1D total variation denoising problem because analysis of problem (10) in the literature is still fairly limited and a closed-form solution even for a restricted family of 2D signals is not available. Still, analysis in 1D can provide practical insights.

A total variation denoising problem can be written in 1D as

$$\hat{u}[x] = \arg \min_u \frac{1}{2} \sum_{x=-L_1+1}^{L_2-1} (u[x] - f[x])^2 + \lambda \sum_{x=-L_1}^{L_2-1} |u[x+1] - u[x]|, \quad (18)$$

where $u \in [-L_1, L_2]$ and $f \in [-L_1 + 1, L_2 - 1]$. For a successful convergence of the AM algorithm it is desirable to have a solution equal or close to the true sharp signal u^0 . By exploiting recent work of Condat [25], Strong and Chan [26] and the *taut string algorithm* of Davies and Kovacs [27], for a simple class of signals, in the following proposition we give the analytical expression for λ that gives a sharp, but scaled and shifted, version of u^0 as the solution of the denoising problem (18).

Proposition 4.2: Let u_0 be a 1D step function of the following form

$$u_0[x] = \begin{cases} U_1 & x \in [-L_1, -1] \\ U_2 & x \in [0, L_2] \end{cases} \quad (19)$$

for some $U_1 < U_2$ and $L_1, L_2 > 2$, and k_0 be a 3-element blur kernel where $\delta_1 \doteq k_0[1]$, $\delta_2 \doteq k_0[-1]$ and $k_0[0] = 1 - \delta_1 - \delta_2$, $\delta_1 + \delta_2 \leq 1$ and $\delta_1, \delta_2 \geq 0$. Then, f is the convolution of u with k

$$f[x] = \begin{cases} U_1 & x \in [-L_1 + 1, -2] \\ U_1 + \delta_1(U_2 - U_1) & x = -1 \\ U_2 - \delta_2(U_2 - U_1) & x = 0 \\ U_2 & x \in [1, L_2 - 1]. \end{cases} \quad (20)$$

The solution $\hat{u}[x]$ to the problem (18) is

$$\hat{u}[x] = \begin{cases} \hat{U}_1(\lambda) & x \in [-L_1, -2] \\ \hat{U}_2(\lambda) & x \in [-1, L_2] \end{cases} \quad (21)$$

TABLE 1
Formulas of \hat{U}_1 and \hat{U}_2 for $\lambda \in [\lambda_{min}^l, \lambda_{max}^l)$, $\lambda \in [\lambda_{min}^c, \lambda_{max}^c)$ and $\lambda \in [\lambda_{min}^r, \lambda_{max}^r)$ used in proposition 4.2.

Regularization domain	$\hat{U}_1(\lambda)$	$\hat{U}_2(\lambda)$
$\lambda \in [\lambda_{min}^l, \lambda_{max}^l)$ $\lambda_{min}^l = (U_2 - U_1)(L_2 - L_2\delta_1 - \delta_2)$ $\lambda_{max}^l = (U_2 - U_1) \frac{L_1-2}{L_1+L_2-1} (L_2 + \delta_1 - \delta_2)$	$U_1 + \frac{\lambda}{L_1-2}$	$\frac{U_1+U_2L_2}{L_2+1} + \frac{(\delta_1-\delta_2)(U_2-U_1)-\lambda}{L_2+1}$
$\lambda \in [\lambda_{min}^c, \lambda_{max}^c)$ $\lambda_{min}^c = (U_2 - U_1) \max\{(L_1 - 2)\delta_1, (L_2 - 1)\delta_2\}$ $\lambda_{max}^c = (U_2 - U_1) \frac{L_2(L_1-1)-L_2\delta_1-(L_1-1)\delta_2}{L_1+L_2-1}$	$U_1 + \frac{\delta_1(U_2-U_1)+\lambda}{L_1-1}$	$U_2 + \frac{-\delta_2(U_2-U_1)-\lambda}{L_2}$
$\lambda \in [\lambda_{min}^r, \lambda_{max}^r)$ $\lambda_{min}^r = (U_2 - U_1)(L_1 - \delta_1 - (L_1 - 1)\delta_2 - 1)$ $\lambda_{max}^r = (U_2 - U_1) \frac{L_2-1}{L_1+L_2-1} (L_1 - \delta_1 + \delta_2 - 1)$	$\frac{U_1(L_1-1)+U_2+(\delta_1-\delta_2)(U_2-U_1)+\lambda}{L_1}$	$U_2 - \frac{\lambda}{L_2-1}$

when $\lambda \in [\lambda_{min}^l, \lambda_{max}^l)$, is

$$\hat{u}[x] = \begin{cases} \hat{U}_1(\lambda) & x \in [-L_1, -1] \\ \hat{U}_2(\lambda) & x \in [0, L_2] \end{cases} \quad (22)$$

when $\lambda \in [\lambda_{min}^c, \lambda_{max}^c)$, and is

$$\hat{u}[x] = \begin{cases} \hat{U}_1(\lambda) & x \in [-L_1, 0] \\ \hat{U}_2(\lambda) & x \in [1, L_2] \end{cases} \quad (23)$$

when $\lambda \in [\lambda_{min}^r, \lambda_{max}^r)$. Analytical expressions for $\hat{U}_1(\lambda)$, $\hat{U}_2(\lambda)$, λ_{min}^l , λ_{max}^l , λ_{min}^c , λ_{max}^c , λ_{min}^r , λ_{max}^r , are defined in Table 1. If $\delta_2 = \frac{L_1-\delta_1-1}{L_1+L_2-2}$ or $\delta_2 = L_2 - (L_1 + L_2 - 2)\delta_1$ then a λ that gives the solution (21), (22) or (23) does not exist.

Proof: See Section A.1. \square

Proposition 4.2 shows that for a wide range of signals it is possible to obtain a sharp signal by solving problem (18). Notice, however, that total variation regularization shifts vertically and locally scales the input signal (see also illustration in Fig 2 (a)). We will show in the next sections that the apparently insignificant scaling has a fundamental role in the convergence of the AM algorithm.

4.2.2 Filtered Image Model

A common practice in many MAP_k methods that use a Variational Bayesian approach is to solve problem (10) using the gradients of u and f instead of the original signals [1], [18], [6]. We consider this case for the AM algorithm and show that it returns again a scaled version of the true sharp signal. The 1D denoising problem of (18) becomes

$$\hat{u}_x[x] = \arg \min_{u_x} \frac{1}{2} \sum_{x=-L_1+1}^{L_2-2} (u_x[x] - f_x[x])^2 + \lambda \sum_{x=-L_1}^{L_2-1} |u_x[x]|. \quad (24)$$

where $u_x[x] = u[x+1] - u[x]$ and $f_x[x] = f[x+1] - f[x]$. Problem (24) can be easily solved in closed form by soft-thresholding, therefore for a simple class of signals as in Proposition 4.2 we can seek for values of λ such that the solution \hat{u}_x is the derivative of a sharp signal, i.e., a Dirac delta.

Proposition 4.3: The solution of problem (24) with $f_x[x] = f[x+1] - f[x]$, where $f[x]$ is defined in (20), is

$$u_x[x] = \begin{cases} 0 & x \neq -2 \\ (\delta_1 - \max(\delta_2, 1 - \delta_1 - \delta_2))(U_2 - U_1) & x = -2 \end{cases} \quad (25)$$

if $\delta_1 > \max(\delta_2, \frac{1-\delta_2}{2})$ and $\lambda \geq (U_2 - U_1) \max(\delta_2, 1 - \delta_1 - \delta_2)$, is

$$u_x[x] = \begin{cases} 0 & x \neq -1 \\ (1 - \delta_1 - \delta_2 - \max(\delta_1, \delta_2))(U_2 - U_1) & x = -1 \end{cases} \quad (26)$$

if $1 > \max(2\delta_1 + \delta_2, 2\delta_2 + \delta_1)$ and $\lambda \geq (U_2 - U_1) \max(\delta_1, \delta_2)$, and is

$$u_x[x] = \begin{cases} 0 & x \neq 0 \\ (\delta_2 - \max(\delta_1, 1 - \delta_1 - \delta_2))(U_2 - U_1) & x = 0 \end{cases} \quad (27)$$

if $\delta_2 > \max(\delta_1, \frac{1-\delta_1}{2})$ and $\lambda \geq (U_2 - U_1) \max(\delta_1, 1 - \delta_1 - \delta_2)$.

If $\delta_2 = \delta_1 \geq 1/3$, if $\delta_1 = (1 - \delta_2)/2 \geq 1/3$ or if $\delta_2 = (1 - \delta_1)/2 \geq 1/3$ then solving problem (24) can never lead to the solutions in (25), (26) and (27) for any possible value of λ .

Proof: See Section A.2. \square

Proposition 4.3 gives conditions on λ such that the TV denoising of filtered signals gives a scaled version of the true sharp signal. This result is similar to the one given by Proposition 4.2 for the classical TV denoising algorithm.

4.2.3 Analysis of the AM Algorithm

The study of the total variation denoising algorithm is important because it represents the first step of the AM algorithm, when the blur is initialized with a Dirac delta. In the next proposition we show how the exact AM algorithm can not leave the no-blur solution (f, δ) when f is defined as in (20) and the first step of the AM algorithm gives a sharp signal as solution.

Proposition 4.4: Let f , u^0 and k^0 be the same as in proposition 4.2. Then, for a $\lambda \in [\lambda_{min}^l, \lambda_{max}^l)$, $\lambda \in [\lambda_{min}^c, \lambda_{max}^c)$ or $\lambda \in [\lambda_{min}^r, \lambda_{max}^r)$ the AM algorithm converges to the solution $k = \delta$ (see Table 1).

Proof: See Section A.3. \square

The result in the above proposition confirms the conclusions of proposition 4.1 analytically for a 1D signal. In section 4.3.1 we also show experimental evidence of the same conclusions.

4.3 The Projected AM Algorithm

Many methods in the literature, and in particular [8], minimize problem (10) by using a variant of the AM algorithm. This variant consists in alternating between minimizing the unconstrained convex problem (15) in u as in the AM algorithm,

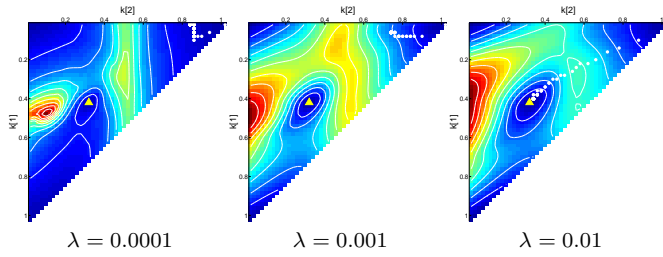


Fig. 3. Illustration of proposition 4.1 (best viewed in color). In this example we show a 1D experiment where we blur a step function with $k_0 = [0.4; 0.3; 0.3]$. We visualize the cost function of eq. (10) for three different values of the parameter λ . Since the blur integrates to 1, only two of the three components are free to take values on a triangular domain (the upper-left triangle in each image). We denote with a yellow triangle the true blur k_0 and with white dots the intermediate blurs estimated during the minimization via the PAM algorithm. Blue pixels have lower values than the red pixels. Dirac delta blurs are located at the three corners of each triangle. At these locations, as well as at the true blur, there are local minima. Notice how the path of the estimated blur on the rightmost image ascends and then descends a hill in the cost functional.

and an unconstrained convex problem in k

$$k^{t+1/3} \leftarrow \arg \min_k \sum_{\mathbf{x} \in \mathcal{F}} ((k \circ u)[\mathbf{x}] - f[\mathbf{x}])^2, \quad (28)$$

followed by two sequential projections where one applies the constraints on the blur k , i.e.,

$$k^{t+2/3} \leftarrow \max\{k^{t+1/3}, 0\}, \quad k^{t+1} \leftarrow \frac{k^{t+2/3}}{\|k^{t+2/3}\|_1}. \quad (29)$$

To distinguish it from the AM, we call this iterative procedure the PAM algorithm. The choice of imposing the constraints sequentially rather than during the gradient descent on k seems a rather unimportant and acceptable approximation of the correct procedure (AM). However, this is not the case and we will see that with this arrangement one can achieve the desired solution.

4.3.1 Analysis of the PAM Algorithm

Our first claim is that this procedure does not minimize the original problem (6). To support this claim we start by showing some experimental evidence in Fig. 3. In this test we work on a 1D version of the problem. We blur a hat function with one blur of size 3 pixels, and we show the minimum of eq. (10) for all possible feasible blurs. Since the blur has only 3 nonnegative components and must add up to 1, we only have 2 free parameters bound between 0 and 1. Thus, we can produce a 2D plot of the minimum of the energy with respect to u as a function of these two parameters. The blue color denotes a small cost, while the red color denotes a large cost. The figures reveal three local minima at the corners, due to the 3 different shifted versions of the no-blur solution, and the local minimum at the true solution ($k_0 = [0.4, 0.3, 0.3]$) marked with a yellow triangle. We also show with white dots

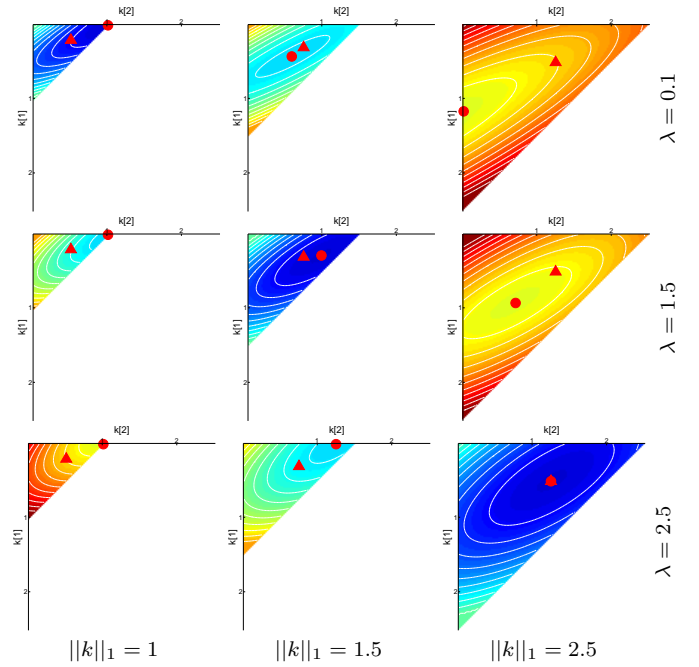


Fig. 4. Illustration of proposition 4.5 (best viewed in color). Each row represents the visualization of the cost function for a particular value of the parameter λ . Each column shows the cost function for three different blur normalizations: $\|k\|_1 = 1, 1.5$, and 2.5 . We denote the scaled true blur $k_0 = [0.2, 0.5, 0.3]$ (with $\|k\|_1 = 1$) with a red triangle and with a red dot the cost function minimum. The color coding is such that: blue < yellow < red; each row shares the same color coding for cross comparison.

the path followed by k estimated via the PAM algorithm by starting from one of the no-blur solutions (upper-right corner). Clearly one can see that the PAM algorithm does not follow a minimizing path in the space of solutions of problem (10). Nonetheless, this algorithm can quite surprisingly succeed in estimating the true blur in two steps even when initialized at the no-blur solution. We provide a proof of this property in the following proposition.

Proposition 4.5: Let f, u^0 and k^0 be the same as in Proposition 4.2. Let also constraint u^0 to be a zero-mean signal. Then, if there exists a $\lambda \in [\lambda_{min}^l, \lambda_{max}^l)$, $\lambda \in [\lambda_{min}^c, \lambda_{max}^c)$ or $\lambda \in [\lambda_{min}^r, \lambda_{max}^r)$ the PAM algorithm estimates the true blur $k = k_0$ (or a shifted version of it) in two steps, when starting from the no-blur solution pair (f, δ) .

Proof: See Appendix A.4. \square

4.4 Extension Beyond the 1D Step Signal

In the discrete settings, we can generalize the analysis developed so far to a piecewise constant signal by dividing it in many step signals. The denoising problem of a piecewise constant signal is then equivalent to many denoising problems of different step signals with careful treatment at the boundaries between each step. One advantage of working with a general piecewise constant signal is that it has better chances than a single step of being correctly reconstructed from its blurry version.

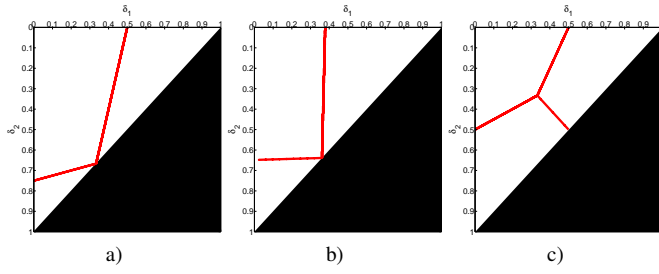


Fig. 5. Illustration of configurations of δ_1 and δ_2 for which it is not possible to estimate a sharp signal in problem (18) (cases a) and b)) and in problem (24) (case c)). The region of feasible solutions is where $\delta_1 + \delta_2 \leq 1$ is satisfied and is denoted by the white region; the red lines denote the configurations that can not lead to a sharp signal for: a) a signal as in proposition 4.2 such that $L_1 = 3$ and $L_2 = 3$; b) a signal as in proposition 4.2 such that $L_1 = 15$ and $L_2 = 24$; c) a filtered signal as in proposition 4.3.

In the case of a 1D step signal, both proposition 4.3 and proposition 4.2 give conditions such that there are blurs that do not allow the reconstruction of a sharp signal. Fortunately, this family of blurs is a set of measure zero in the space of all feasible blurs. We illustrate this set in Fig. 5 for the space of all possible 3-element blurs parametrized in δ_1 and δ_2 and call it *degenerate set*. In the classical TV denoising algorithm from proposition 4.3, the degenerate set consists of two lines, parametrized in L_1 and L_2 (see Fig. 5 (a) and (b)). Because of the parametrization on L_1 and L_2 , different 1D step signals have different degenerate sets. However, in a piecewise constant signal each step will have a different degenerate set. Thus, it is likely that for each blur configuration, some steps of the signal will not be in a degenerate case and allow the reconstruction of the whole signal.

For the TV denoising of the filtered 1D step signal, the degenerate set is denoted by three segments (see Fig. 5 (c)). Since this set is the same for any possible 1D step signal, the likelihood of reconstructing a sharp signal would not change when using a piecewise constant signal. The discussion on filtered signals is not developed further due to space constraints. Still, the above argument suggests that there may be aspects, previously not considered in the literature, that support the use of the original signal in problem (10) in lieu of their filtered versions.

4.5 Discussion

Propositions (4.5) and (4.4) show that with 1D zero-mean step signals and no-blur initialization, for some values of λ PAM converges to the correct blur (and only in 2 steps) while AM does not. We used a step signal to keep the discussion simple and because it is fundamental to illustrate the behavior of both algorithms at edges in a 2D image. However, as already suggested in sec. 4.4, extensions of the above results to blurs with a larger support and beyond step functions are possible by dividing the signal in many step functions and by a careful treatment of the boundaries.

In Fig. 4 we illustrate two further aspects of proposition 4.5 (it is recommended that images are viewed in color): 1) the advantage of a non unitary normalization of k during the optimization step (which is a key feature of PAM) and 2) the need for a sufficiently large regularization parameter λ . In the top row images we set $\lambda = 0.1$. Then, we show the cost $\|k * u^1 - f\|_2^2$, with $u^1 = \arg \min_u \|u - f\|_2^2 + \lambda J(u)$, for all possible 1D blurs k with a 3-pixel support under the constraints $\|k\|_1 = 1, 1.5, 2.5$ respectively. This is the cost that PAM minimizes at the second step when initialized with $k = \delta$. Because k has three components and we fix its normalization, we can illustrate the cost as a 2D function of $k[1]$ and $k[2]$ as in Fig. 3. However, as the normalization of k grows, the triangular domain of $k[1]$ and $k[2]$ increases as well. Since the optimization of the blur k is unconstrained, the optimal solution will be searched both within the domain and across normalization factors. Thanks to the color coding scheme, one can immediately see that the case of $\|k\| = 1$ achieves the global minimum, and hence the solution is the Dirac delta. However, as we set $\lambda = 1.5$ in the second row or $\lambda = 2.5$ in the bottom row, we can see a shift of the optimal value for non unitary blur normalization values and also for a shift of the global minimum to the desired solution (bottom-right plot).

5 IMPLEMENTATION

In Algorithm 1 we show the pseudo-code of our adaptation of the PAM algorithm. At each iteration we perform just one gradient descent step on u and on k as proposed by Chan and Wong [8] because we experimentally noticed that this speeds up the convergence of the algorithm. The gradient descent updates the sharp image u at the t -th iteration with

$$u^{t+1} \leftarrow u^t - \epsilon_u \left(k_-^t \bullet (k^t \circ u^t - f) - \lambda \nabla \cdot \frac{\nabla u^t}{|\nabla u^t|} \right) \quad (30)$$

for some step $\epsilon > 0$ where $k_-(x) = k(-x)$ and \bullet denotes the discrete convolution operator where the result is the full convolution region, i.e., if $f = u \bullet k$, with $k \in \mathbf{R}^{h \times w}$, $u \in \mathbf{R}^{m \times n}$, then we have $f \in \mathbf{R}^{(m+h-1) \times (n+w-1)}$ with zero padding as boundary condition. The iteration on the blur kernel k is instead given by

$$k^{t+1/3} \leftarrow k^t - \epsilon_k (u_-^{t+1} \circ (k^t \circ u^{t+1} - f)). \quad (31)$$

From proposition (4.5) we know that a big value for the parameter λ helps avoiding the no-blur solution, but in practice it also makes the estimated sharp image u too “cartooned”. We found that iteratively reducing the value of λ as specified in Algorithm 1 helps getting closer to the true solution. In the following paragraphs we highlight some other important features of the PAM algorithm.

Pyramid scheme. While all the theory holds at the original input image size, to speed up the algorithm we also make use of a pyramid scheme, where we scale down the blurry image and the blur size until the latter is 3×3 pixels. We then launch our deblurring algorithm from the lowest scale, then upsample the results and use them as initialization for the following scale. This procedure provides a significant speed up of the algorithm. On the smallest scale, we initialize our

Data: f , size of blur, initial large λ , final λ_{min}

Result: u, k

```

1  $u_0 \leftarrow \text{pad}(f)$ ;
2  $k_0 \leftarrow \text{uniform}$ ;
3 while not converged do
4    $u^{t+1} \leftarrow u^t - \epsilon_u \left( k^t \bullet (k^t \circ u^t - f) - \lambda \nabla \cdot \frac{\nabla u^t}{|\nabla u^t|} \right)$ ;
5    $k^{t+1/3} \leftarrow k^t - \epsilon_k (u^{t+1} \circ (k^t \circ u^{t+1} - f))$ ;
6    $k^{t+2/3} \leftarrow \max\{k^{t+1/3}, 0\}$ ;
7    $k^{t+1} \leftarrow \frac{k^{t+2/3}}{\|k^{t+2/3}\|_1}$ ;
8    $\lambda \leftarrow \max\{0.99\lambda, \lambda_{min}\}$ ;
9    $t \leftarrow t + 1$ ;
10 end
11  $u \leftarrow u^{t+1}$ ;
12  $k \leftarrow k^{t+1}$ ;

```

Algorithm 1: Blind Deconvolution PAM algorithm

optimization from a uniform blur.

Color images. For the blur estimation many methods first convert color images to grayscale. In contrast with this common practice, we extended our algorithm to work directly with color images. Recently, many papers have proposed algorithms for color denoising using TV regularization [28], [29], [30]. In our implementation we use the method of Blomgren and Chan [30], and all the results on color images that we show in this papers are obtained by solving the blind-deconvolution problem on the original color space.

6 EXPERIMENTS

In this section we show empirically how a straightforward implementation of the simple PAM algorithm is able to compete with more sophisticated algorithms. We provide an unoptimized Matlab implementation of the PAM algorithm on our website⁵. Our implementation processes images of 255×225 pixels with blurs of about 20×20 pixels in around 2-5 minutes, while our non-blind deconvolution algorithm takes about 10 – 30 seconds.

In a first set of experiments we used the dataset from [11] in the same manner illustrated by the authors. For the whole test we used $\lambda_{min} = 0.0006$. We used the non-blind deconvolution algorithm from [31] with $\lambda = 0.0068$ and for each method we compute the cumulative histogram of the deconvolution error ratio across test examples, where the i -th bin counts the percentage of images for which the error ratio is smaller than i . The deconvolution error ratio, as defined in [11], measures the ratio between the SSD deconvolution error with the estimated and correct kernels. In Fig. 8 we show the cumulative histogram of the error ratio of the PAM algorithm compared to several state-of-the-art algorithms. The PAM algorithm performs on par with the one from Levin *et al.* [31], with a slightly higher number of restored images with small error ratio.

In a second set of experiments we used the dataset from [32]. The evaluation has been performed similarly to the previous experiment, but we used the non-blind deconvolution algorithm of Zoran and Weiss [33] as proposed in [32]. In this case we used $\lambda_{min} = 0.001$. The PAM algorithm

performs better than very recent algorithms that exploit sophisticated patch-based priors such as the ones from Michaeli and Irani [34] and Sun *et al.* [32]. The algorithm of Perrone *et al.* [35] is however able to achieve a better performance.

Recently Köhler *et al.* [36] introduced a dataset of images blurred by camera shake blur. Even if this kind of artifact produces space-varying blur, many algorithms that assume shift-invariant blur perform well on many images of the dataset. In Fig. 12 we show an example of reconstructed image from the dataset of [36] where the PAM algorithm is as robust to camera shake blur as other shift-invariant algorithms.

In Fig. 10 we show a comparison between our PAM implementation and the one proposed by Xu and Jia [5]. Their algorithm is able to restore sharp images even when the blur size is large by using an edge selection scheme that selects only large edges. This behavior is automatically mimicked by the PAM algorithm thanks to the TV prior. Also, in the presence of noise, the PAM algorithm performs visually on a par with the state-of-the-art algorithms as shown in Fig. 11.

For testing purposes we also adapted our algorithm to support different boundary conditions by substituting the convolution operator described in (9) with the one in (8).

In Fig. 6 we show a comparison on the dataset of [11] between our original approach and the adaptations with different boundary conditions. For each boundary condition we evaluated the ratios between the SSD deconvolution errors of the estimated and correct kernels. The implementations with the different boundary conditions perform worse than our free-boundary implementation, even if pre-processing the blurry image with the method of Liu and Jia [23] considerably improves the performance of the *periodic* boundary condition.

Recent algorithms estimate the blur by using filtered versions of u and f in the data fitting term (typically the gradients or the Laplacian). This choice might improve the estimation of the blur because it reduces the error at the boundary when using any of the previous approximations, but it might result also in a larger sensitivity to noise. In Fig. 7 we show how with the use of the filtered images for the blur estimation the performance of the *periodic* and *replicate* boundary conditions improves, while the others get slightly worse. Notice that our implementation still achieves better results than other boundary assumptions.

7 CONCLUSIONS

In this paper we shed light on approaches to solve blind deconvolution. First, we confirmed that the problem formulation of total variation blind deconvolution as a maximum a priori in both sharp image and blur ($\text{MAP}_{u,k}$) is prone to local minima and, more importantly, does not favor the correct solution. Second, we also confirmed that the original implementation [8] of total variation blind deconvolution (PAM) can successfully achieve the desired solution. This discordance was clarified by dissecting PAM in its simplest steps. The analysis revealed that such algorithm does not minimize the original $\text{MAP}_{u,k}$ formulation. This analysis applies to a large number of methods solving $\text{MAP}_{u,k}$ as they might exploit the properties of PAM; moreover, it shows that there might be

5. <http://www.cvg.unibe.ch/dperrone/tvdb/>

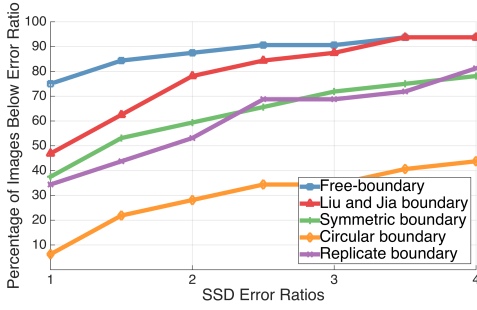


Fig. 6. Comparison of PAM algorithm with different boundary conditions.

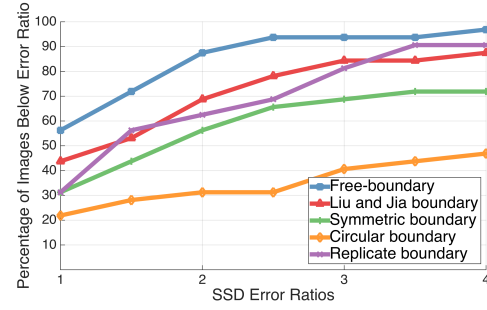


Fig. 7. As in Fig. 6, but using a filtered version of the images for the blur estimation.

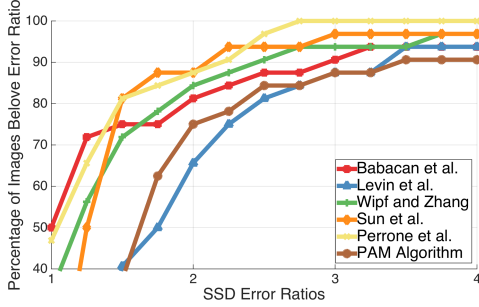


Fig. 8. Comparison between the PAM algorithm and recent state-of-the-art algorithms on the dataset [37].

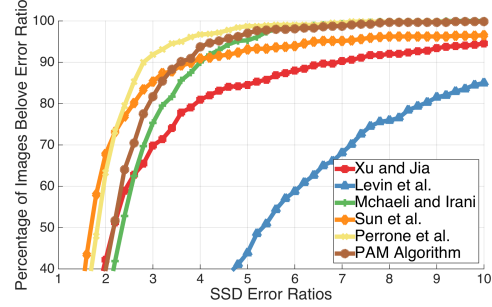


Fig. 9. Comparison between the PAM algorithm and recent state-of-the-art algorithms on the dataset [32].



Blurry Input.



Restored image and blur with Xu and Jia [5].



Restored image and blur with PAM algorithm.

Fig. 10. Example of blind-deconvolution image and blur (bottom-right insert) restoration.

principled solutions to $\text{MAP}_{u,k}$. We believe that by further studying the behavior of the PAM algorithm one could arrive at novel useful formulations for blind deconvolution. Finally, we have showed that the PAM algorithm is neither worse nor better than the state of the art algorithms despite its simplicity.

APPENDIX A

A.1 Proof of Proposition 4.2

Proof: Solving problem (18) is equivalent to solving the following problem

$$\bar{u}[x] = \arg \min_u \frac{1}{2} \sum_{x=-L_1+1}^{L_2-1} (u[x] - f[x])^2 + \lambda \sum_{x=-L_1+1}^{L_2-2} |u[x+1] - u[x]|. \quad (32)$$

where $\bar{u}[x] = \hat{u}[x]$ for $x \in [-L_1+1, L_2-1]$, $\hat{u}[-L_1] = \hat{u}[-L_1+1]$ and $\hat{u}[L_2] = \hat{u}[L_2-1]$. In the following, unless specified, we will always refer to problem (32).

The solution of problem (32) can also be written as $\bar{u}[x] = \hat{s}[x] - \hat{s}[x-1]$, $x \in [-L_1+1, L_2-1]$, where \hat{s} is found by solving the *taut string* problem (e.g., see Davies and Kovac [27])

$$\begin{aligned} \hat{s}[x] &= \arg \min_s \sum_{x=-L_1+1}^{L_2-1} \sqrt{1 + |s[x] - s[x-1]|^2} \\ \text{s.t.} \quad &\max_{x \in [-L_1+1, L_2-1]} |s[x] - r[x]| \leq \lambda \quad \text{and} \\ &s[-L_1] = 0, \quad s[L_2-1] = r[L_2-1] \end{aligned} \quad (33)$$

where $r[x] = \sum_{y=-L_1+1}^x f[y]$ with $x \in [-L_1+1, L_2-1]$. Given the explicit form of f in eq. (20) we obtain that

$$r[x] = \begin{cases} U_1(x+L_1) & x \in [-L_1+1, -2] \\ U_1(L_1-1) + \delta_1(U_2 - U_1) & x = -1 \\ U_1(L_1-1) + U_2(x+1) + (\delta_1 - \delta_2)(U_2 - U_1) & x \in [0, L_2-1]. \end{cases} \quad (34)$$

Notice that $r[x]$ has three discontinuities at $x = -2$, $x = -1$ and $x = 0$. Let consider solving the taut string problem by enforcing in turn only the constraint $|s[-2] - r[-2]| \leq \lambda$, $|s[-1] - r[-1]| \leq \lambda$ and $|s[0] - r[0]| \leq \lambda$.

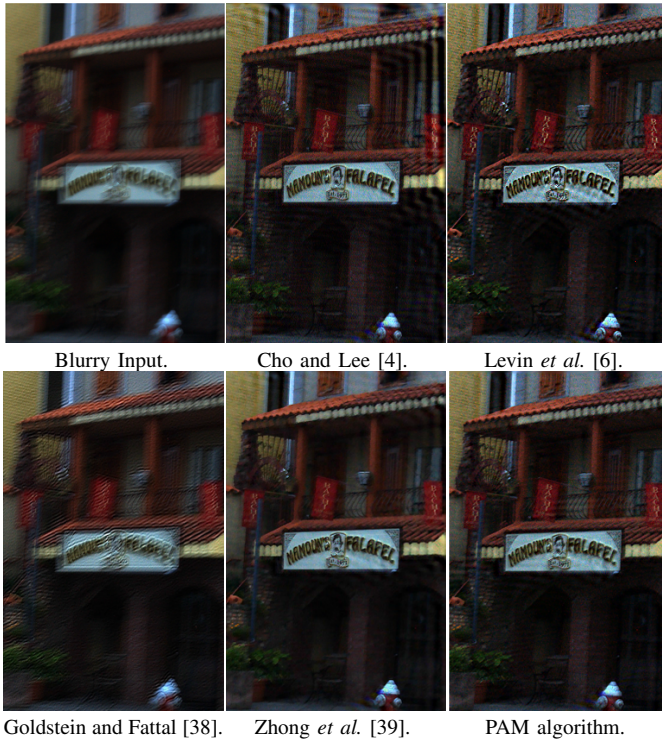


Fig. 11. Examples of blind-deconvolution restoration.

For the first case the cost of the taut string problem is minimum for the shortest path s through a point at $x = -2$. We can decompose such path into the concatenation of the shortest path from $x = -L_1 + 1$ to $x = -2$ and the shortest path from $x = -2$ to $x = L_2 - 1$. Given that each of these paths are only constrained at the end points, a direct solution will give a line segment between the end points, i.e.,

$$s[x] = \begin{cases} \frac{x+L_1}{L_1-2} s[-2] & x \in [-L_1 + 1, -2] \\ \frac{L_2-1-x}{L_2-1} s[-2] + \frac{x+2}{L_2+1} r[L_2 - 1] & x \in [-1, L_2 - 1]. \end{cases} \quad (35)$$

The value $s[-2]$ that yields the shortest path and satisfies the constraint

$$r[-2] - \lambda \leq s[-2] \leq r[-2] + \lambda \quad (36)$$

is $s[-2] = U_1(L_1 - 2) + \lambda$ when $\lambda < (U_2 - U_1) \frac{L_1-2}{L_1+L_2-1} (L_2 + \delta_1 - \delta_2)$ and $s[-2] = U_1(L_1 - 2) + (U_2 - U_1) \frac{L_1-2}{L_1+L_2-1} (L_2 + \delta_1 - \delta_2)$ otherwise.

Now, we will show that, given $\lambda \geq (U_2 - U_1)(L_2 - L_2\delta_1 - \delta_2)$, the above shortest path s is also the solution \hat{s} to the taut string problem (33) with all the constraints. It will suffice to show that this path satisfies all the constraints in the taut string problem. Then, since it is the shortest path with a single constraint, it must also be the shortest path for problem (33). To verify all the constraints, we only need to consider 2 cases:

$$\begin{aligned} x = -1 &\rightarrow \frac{|(L_2 - L_2\delta_1 - \delta_2)(U_2 - U_1) + \lambda L_2|}{\lambda(L_2 + 1)} \leq 1 \\ x = 0 &\rightarrow \frac{|(L_2 - 1)((1 - \delta_1 + \delta_2)(U_2 - U_1) + \lambda)|}{\lambda(L_2 + 1)} \leq 1 \end{aligned} \quad (37)$$

as all the others are directly satisfied when these are. The first inequality leads to the condition $\lambda \geq (U_2 - U_1)(L_2 - L_2\delta_1 - \delta_2)$ and the second inequality to the condition $\lambda \geq (U_2 - U_1) \frac{L_2-1}{2} (1 - \delta_1 + \delta_2)$. However, as long as $L_1, L_2 > 2$, $(U_2 - U_1)(L_2 - L_2\delta_1 - \delta_2) \geq (U_2 - U_1) \frac{L_2-1}{2} (1 - \delta_1 + \delta_2)$ for $\delta_1 + \delta_2 \leq 1$, therefore it is sufficient to have the condition $\lambda \geq (U_2 - U_1)(L_2 - L_2\delta_1 - \delta_2)$ to satisfy both inequalities. We can then obtain $\hat{u}[x] = \hat{s}[x] - \hat{s}[x - 1]$ from

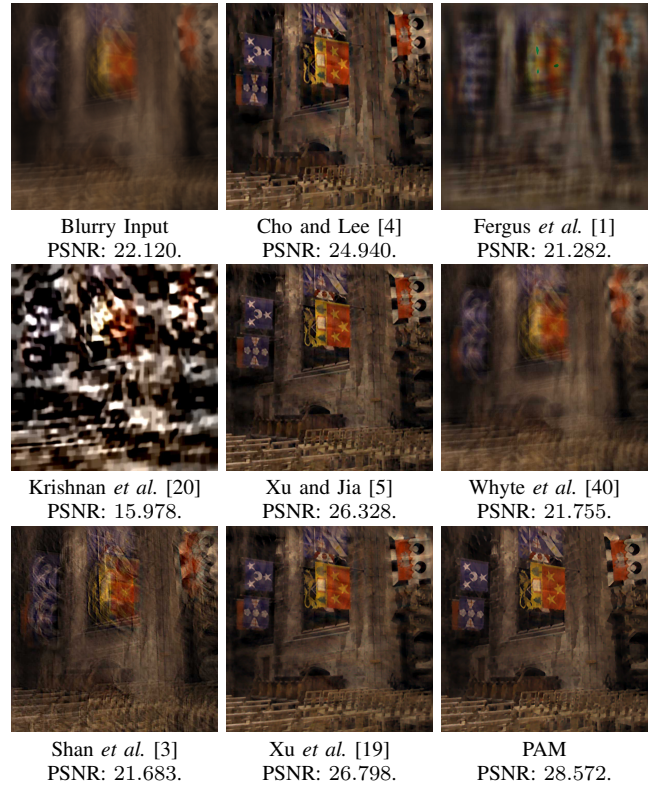


Fig. 12. Examples of blind-deconvolution image reconstructions from dataset [36].

eq. (43) and write

$$\hat{u}[x] = \begin{cases} U_1 + \frac{\lambda}{L_1-2} & x \in [-L_1 + 1, -2] \\ \frac{U_1 + U_2 L_2}{L_2 + 1} + \frac{(\delta_1 - \delta_2)(U_2 - U_1) - \lambda}{L_2 + 1} & x \in [-1, L_2 - 1] \end{cases} \quad (38)$$

If $\frac{L_1-2}{L_1+L_2-1} (L_2 + \delta_1 - \delta_2) \leq L_2 - L_2\delta_1 - \delta_2$ the two conditions on λ mentioned above can never be satisfied. With some algebraic manipulation we obtain that if $\delta_2 \leq L_2 - (L_1 + L_2 - 2)\delta_2$ a λ such that the solution (38) is obtained does not exist.

In a similar manner we can consider the second point $x = -1$ and solve the taut string problem imposing only the constraint $|s[-1] - r[-1]| \leq \lambda$. Also in this case we can decompose the shortest path s into the concatenation of the shortest path from $x = -L_1 + 1$ to $x = -1$ and then one from $x = -1$ to $x = L_2 - 1$. A direct solution will give a line segment between the end points,

$$s[x] = \begin{cases} \frac{x+L_1}{L_1-1} s[-1] & x \in [-L_1 + 1, -1] \\ \frac{L_2-1-x}{L_2} s[-1] + \frac{x+1}{L_2} r[L_2 - 1] & x \in [0, L_2 - 1]. \end{cases} \quad (39)$$

The value $s[-1]$ that yields the shortest path and satisfies the constraint

$$r[-1] - \lambda \leq s[-1] \leq r[-1] + \lambda \quad (40)$$

is $s[-1] = U_1(L_1 - 1) + \delta_1(U_2 - U_1) + \lambda$ when $\lambda < (U_2 - U_1) \frac{L_2(L_1-1) - L_2\delta_1 - (L_1-1)\delta_2}{L_1+L_2-1}$ and $s[-1] = U_1(L_1 - 1) + \delta_1(U_2 - U_1) + (U_2 - U_1) \frac{L_2(L_1-1) - L_2\delta_1 - (L_1-1)\delta_2}{L_1+L_2-1}$ otherwise.

Now, we will show that, given $\lambda \geq (U_2 - U_1) \max\{(L_1 - 2)\delta_1, (L_2 - 1)\delta_2\}$, the above shortest path s is also the solution \hat{s} to the taut string problem (33) with all the constraints. To verify all the constraints, we only need to consider 2 cases:

$$\begin{aligned} x = -2 &\rightarrow |(L_1 - 2)(\delta_1(U_2 - U_1) + \lambda)| \leq \lambda(L_1 - 1) \\ x = 0 &\rightarrow |(L_2 - 1)(\delta_2(U_2 - U_1) + \lambda)| \leq \lambda L_2 \end{aligned} \quad (41)$$

as all the others are directly satisfied when these are. By direct substitution, one can find that $\lambda \geq (U_2 - U_1) \max\{(L_1 - 2)\delta_1, (L_2 - 1)\delta_2\}$ satisfies all the above constraints as long as $L_1, L_2 > 2$. We can then obtain $\bar{u}[x] = \hat{s}[x] - \hat{s}[x-1]$ from eq. (43) and write

$$\hat{u}[x] = \begin{cases} U_1 + \frac{\delta_1(U_2 - U_1) + \lambda}{L_1 - 1} & x \in [-L_1, -1] \\ U_2 + \frac{-\delta_2(U_2 - U_1) - \lambda}{L_2} & x \in [0, L_2] \end{cases} \quad (42)$$

Also for this case there are configurations of L_1, L_2, δ_1 and δ_2 for which a λ that gives (42) does not exist. We distinguish two cases: if $(L_2 - 1)\delta_2 < (L_1 - 2)\delta_1$ then the condition $\frac{L_2(L_1 - 1) - L_2\delta_1 - (L_1 - 1)\delta_2}{L_1 + L_2 - 1} \leq (L_2 - 1)\delta_2$ must be satisfied; or, if $(L_2 - 1)\delta_2 \geq (L_1 - 2)\delta_1$, the condition $\frac{L_2(L_1 - 1) - L_2\delta_1 - (L_1 - 1)\delta_2}{L_1 + L_2 - 1} \leq (L_2 - 1)\delta_2$ must be satisfied. With simple algebraic manipulation we have for the first case if $\delta_2 < \frac{(L_1 - 2)\delta_1}{L_2 - 1}$ that $\delta_2 \geq L_2 - (L_1 + L_2 - 2)\delta_1$, or if $\delta_2 \geq \frac{(L_1 - 2)\delta_1}{L_2 - 1}$ that $\delta_2 > \frac{L_1 - \delta_1 - 1}{L_1 + L_2 - 2}$ must be satisfied. For the last point $x = 0$ we solve the taut string problem imposing only the constraint $|s[0] - r[0]| \leq \lambda$, having a direct solution as following

$$s[x] = \begin{cases} \frac{x + L_1}{L_1} s[0] & x \in [-L_1 + 1, 0] \\ \frac{L_2 - 1 - x}{L_2 - 1} s[0] + \frac{x}{L_2 - 1} r[L_2 - 1] & x \in [1, L_2 - 1]. \end{cases} \quad (43)$$

The value $s[0]$ that yields the shortest path and satisfies the constraint

$$r[0] - \lambda \leq s[0] \leq r[0] + \lambda \quad (44)$$

is $s[0] = U_2 + (\delta_1 - \delta_2)(U_2 - U_1) + U_1(L_1 - 1) + \lambda$ when $\lambda < (U_2 - U_1) \frac{L_2 - 1}{L_1 + L_2 - 1} (L_1 - \delta_1 + \delta_2 - 1)$ and $s[-1] = U_2 + (\delta_1 - \delta_2)(U_2 - U_1) + U_1(L_1 - 1) + (U_2 - U_1) \frac{L_2 - 1}{L_1 + L_2 - 1} (L_1 - \delta_1 + \delta_2 - 1)$ otherwise.

For $\lambda \geq (U_2 - U_1)(L_1 - \delta_1 - (L_1 - 1)\delta_2 - 1)$, the above shortest path s is also the solution \hat{s} to the taut string problem (33) with all the constraints. Indeed, it satisfies the following 2 cases:

$$\begin{aligned} x = -2 &\rightarrow \frac{|(L_1 - 2)(\delta_1 - \delta_2 + 1)(U_2 - U_1) + (L_2 - 2)\lambda|}{\lambda L_1} \leq 1 \\ x = -1 &\rightarrow \frac{|((L_1 - 1)(1 - \delta_2) - \delta_1)(U_2 - U_1) + (L_1 - 1)\lambda|}{\lambda L_1} \leq 1 \end{aligned} \quad (45)$$

for $L_1, L_2 > 2$ and $\delta_1 + \delta_2 \leq 1$. We can then obtain $\bar{u}[x] = \hat{s}[x] - \hat{s}[x-1]$ from eq. (43) and write

$$\hat{u}[x] = \begin{cases} \frac{U_1(L_1 - 1) + U_2 + (\delta_1 - \delta_2)(U_2 - U_1) + \lambda}{L_1} & x \in [-L_1, 0] \\ U_2 - \frac{\lambda}{L_2 - 1} & x \in [1, L_2] \end{cases} \quad (46)$$

If $\frac{L_2 - 1}{L_1 + L_2 - 1} (L_1 - \delta_1 + \delta_2 - 1) \leq L_1 - \delta_1 - (L_1 - 1)\delta_2 - 1$ the two conditions on λ mentioned above can never be satisfied. With some algebraic manipulation we obtain that if $\delta_2 \leq \frac{(L_1 - 1) - \delta_1}{L_1 + L_2 - 2}$ a λ such that the solution (46) is obtained does not exist.

If the conditions $\delta_2 \leq \frac{L_1 - \delta_1 - 1}{L_1 + L_2 - 2}$, $\delta_2 \leq L_2 - (L_1 + L_2 - 2)\delta_1$, $\delta_2 < \frac{(L_1 - 2)\delta_1}{L_2 - 1}$ and $\delta_2 \geq L_2 - (L_1 + L_2 - 2)\delta_1$ or $\delta_2 \geq \frac{(L_1 - 2)\delta_1}{L_2 - 1}$ and $\delta_2 \geq \frac{L_1 - \delta_1 - 1}{L_1 + L_2 - 2}$ are all satisfied none of the solutions (38), (42) and (46) can be obtained. That is the case only if the conditions $\delta_2 = \frac{L_1 - \delta_1 - 1}{L_1 + L_2 - 2}$ or $\delta_2 = L_2 - (L_1 + L_2 - 2)\delta_1$ are true. \square

A.2 Proof of proposition (4.3)

Proof: From the definition of $f_x[x]$ and $f[x]$ we have

$$f_x[x] = \begin{cases} 0 & x \in [-L_1 + 1, -3] \\ \delta_1(U_2 - U_1) & x = -2 \\ (1 - \delta_1 - \delta_2)(U_2 - U_1) & x = -1 \\ \delta_2(U_2 - U_1) & x = 0 \\ 0 & x \in [1, L_2 - 2] \end{cases} \quad (47)$$

The problem (24) is equivalent to solving

$$\bar{u}_x[x] = \arg \min_{u_x} \frac{1}{2} \sum_{x=-L_1+1}^{L_2-2} (u_x[x] - f_x[x])^2 + \lambda \sum_{x=-L_1+1}^{L_2-2} |u_x[x]|. \quad (48)$$

where $\hat{u}_x[x] = \bar{u}_x[x]$ for $x \in [-L_1 + 1, L_2 - 2]$ and $\hat{u}_x[-L_1] = 0$, $\hat{u}_x[L_2 - 1] = 0$.

The solution of problem (48) can be computed in closed form and it is equal to

$$\bar{u}_x[x] = \max(f_x[x] - \lambda \text{sign}(f_x[x]), 0). \quad (49)$$

It is possible to obtain three different Delta dirac functions using (49).

If $\delta_1 > \max(\delta_2, \frac{1 - \delta_2}{2})$, applying (49) with $\lambda \geq (U_2 - U_1) \max(\delta_1, 1 - \delta_1 - \delta_2)$ would lead to

$$u_x[x] = \begin{cases} 0 & x \neq -2 \\ (\delta_1 - \max(\delta_2, 1 - \delta_1 - \delta_2))(U_2 - U_1) & x = -2 \end{cases} \quad (50)$$

If $1 > \max(2\delta_1 + \delta_2, 2\delta_2 + \delta_1)$, applying (49) with $\lambda \geq (U_2 - U_1) \max(\delta_1, \delta_2)$ would lead to

$$u_x[x] = \begin{cases} 0 & x \neq -1 \\ (1 - \delta_1 - \delta_2 - \max(\delta_1, \delta_2))(U_2 - U_1) & x = -1 \end{cases} \quad (51)$$

If $\delta_2 > \max(\delta_1, \frac{1 - \delta_1}{2})$, applying (49) with $\lambda \geq (U_2 - U_1) \max(\delta_1, 1 - \delta_1 - \delta_2)$ would lead to

$$u_x[x] = \begin{cases} 0 & x \neq 0 \\ (\delta_2 - \max(\delta_1, 1 - \delta_1 - \delta_2))(U_2 - U_1) & x = 0 \end{cases} \quad (52)$$

If none of the conditions $\delta_1 > \max(\delta_2, \frac{1 - \delta_2}{2})$, $1 > \max(2\delta_1 + \delta_2, 2\delta_2 + \delta_1)$ and $\delta_2 > \max(\delta_1, \frac{1 - \delta_1}{2})$ is satisfied then it is not possible to obtain a Dirac delta function from $f_x[x]$ solving problem (48). This corresponds to the conditions $\delta_2 = \delta_1 \geq 1/3$, $\delta_1 = (1 - \delta_2)/2 \geq 1/3$ or $\delta_2 = (1 - \delta_1)/2 \geq 1/3$. \square

A.3 Proof of proposition (4.4)

Proof: In this proof we will show that the cost $\|k * \hat{u} - f\|_2^2$ is minimum for $k = \delta$, where the constraint $\sum_x k[x] = 1$ is enforced and \hat{u} is obtained by solving the first step of the AM algorithm (15) for the given values of λ . To make calculations easier we write k as a 3-element blur kernel where $\hat{\delta}_1 \doteq k[1]$, $\hat{\delta}_2 \doteq k[-1]$ and $k[0] = 1 - \hat{\delta}_1 - \hat{\delta}_2$, $\hat{\delta}_1 + \hat{\delta}_2 \leq 1$ and $\hat{\delta}_1, \hat{\delta}_2 \geq 0$. Notice that in this form the constraint $\sum_x k[x] = 1$ is implicitly enforced.

For a $\lambda \in [\lambda_{min}^c, \lambda_{max}^c]$ from Proposition 4.2 we have that the minimizer of problem (15) is

$$\hat{u}[x] = \begin{cases} \hat{U}_1 & x \in [-L_1, -1] \\ \hat{U}_2 & x \in [0, L_2]. \end{cases} \quad (53)$$

The cost $\|k * \hat{u} - f\|_2^2$, can be then split in 4 regions

$$\begin{aligned} \|k * \hat{u} - f\|_2^2 &= \sum_x ((k * \hat{u})[x] - f[x])^2 = \\ &+ (L_1 - 3)(\hat{U}_1 - U_1)^2 \\ &+ (\hat{U}_1 + \hat{\delta}_1(\hat{U}_2 - \hat{U}_1) - (U_1 + \delta_1(U_2 - U_1)))^2 \\ &+ ((\hat{U}_2 - \hat{\delta}_2(\hat{U}_2 - \hat{U}_1) - (U_2 - \delta_2(U_2 - U_1)))^2 \\ &+ (L_2 - 2)(\hat{U}_2 - U_2)^2 \end{aligned} \quad (54)$$

The first and forth terms do not depend on k , so only the other two terms contribute to the estimation of k . Notice that the inequalities $\hat{U}_1 \geq U_1 + \delta_1(U_2 - U_1) > U_1$, and $\hat{U}_2 \leq U_2 - \delta_2(U_2 - U_1) < U_2$ hold. This means that $\hat{U}_1 - (U_1 + \delta_1(U_2 - U_1))$ is positive, therefore the value of $\hat{\delta}_1 \geq 0$ that minimizes the second term is $\hat{\delta}_1 = 0$, and, because $\hat{U}_2 - (U_2 - \delta_2(U_2 - U_1))$ is negative, the value of $\hat{\delta}_2 \geq 0$ that minimizes the third term is $\hat{\delta}_2 = 0$. This shows that the k that minimizes the cost (54) is the Dirac delta where $k[1] = \hat{\delta}_1 = 0$, $k[-1] = \hat{\delta}_2 = 0$ and $k[0] = 1$.

For $\lambda \in [\lambda_{min}^l, \lambda_{max}^l]$ from Proposition 4.2 we have that the minimizer of problem (15) is

$$\hat{u}[x] = \begin{cases} \hat{U}_1 & x \in [-L_1, -2] \\ \hat{U}_2 & x \in [-1, L_2]. \end{cases} \quad (55)$$

The cost $\|k * \hat{u} - f\|_2^2$, can be then split in 5 regions

$$\begin{aligned} \|k * \hat{u} - f\|_2^2 &= \sum_x ((k * \hat{u})[x] - f[x])^2 = \\ &+ (L_1 - 2)(\hat{U}_1 - U_1)^2 \\ &+ (\hat{U}_1 + \hat{\delta}_1(\hat{U}_2 - \hat{U}_1) - U_1)^2 \\ &+ ((\hat{U}_2 - \hat{\delta}_2(\hat{U}_2 - \hat{U}_1) - (U_1 + \delta_1(U_2 - U_1)))^2 \\ &+ ((\hat{U}_2 - (U_2 - \delta_2(U_2 - U_1)))^2 \\ &+ (L_2 - 2)(\hat{U}_2 - U_2)^2 \end{aligned} \quad (56)$$

The first, forth and fifth terms do not depend on k , so only the other two terms contribute to the estimation of k . Notice that the inequalities $\hat{U}_1 > U_1$ and $\hat{U}_2 \leq U_1 + \delta_1(U_2 - U_1) < U_2 - \delta_2(U_2 - U_1) < U_2$ hold. This means that, because $\hat{U}_1 - U_1$ is positive, the value of $\hat{\delta}_1 \geq 0$ that minimizes the third term is $\hat{\delta}_1 = 0$, and that $\hat{U}_2 - \hat{U}_2 - U_1 + \delta_1(U_2 - U_1)$ is negative, therefore the value of $\hat{\delta}_2 \geq 0$ that minimizes the second term is $\hat{\delta}_2 = 0$. This shows that the k that minimizes the cost (54) is a Dirac delta where $k[1] = \hat{\delta}_1 = 0$, $k[-1] = \hat{\delta}_2 = 0$ and $k[0] = 1$.

For a $\lambda \in [\lambda_{min}^r, \lambda_{max}^r]$ from Proposition 4.2 we have that the minimizer of problem (15) is

$$\hat{u}[x] = \begin{cases} \hat{U}_1 & x \in [-L_1, 0] \\ \hat{U}_2 & x \in [1, L_2]. \end{cases} \quad (57)$$

The cost $\|k * \hat{u} - f\|_2^2$, can be then split in 5 regions

$$\begin{aligned} \|k * \hat{u} - f\|_2^2 &= \sum_x ((k * \hat{u})[x] - f[x])^2 = \\ &+ (L_1 - 3)(\hat{U}_1 - U_1)^2 \\ &+ (\hat{U}_1 - (U_1 + \delta_1(U_2 - U_1)))^2 \\ &+ (\hat{U}_1 + \hat{\delta}_1(\hat{U}_2 - \hat{U}_1) - (U_2 - \delta_2(U_2 - U_1)))^2 \\ &+ ((\hat{U}_2 - \hat{\delta}_2(\hat{U}_2 - \hat{U}_1) - U_2)^2 \\ &+ (L_2 - 3)(\hat{U}_2 - U_2)^2 \end{aligned} \quad (58)$$

The first, second and fifth terms do not depend on k , so only the other two terms contribute to the estimation of k . Notice that the inequalities $\hat{U}_1 \geq U_2 - \delta_2(U_2 - U_1) > U_1 + \delta_1(U_2 - U_1) > U_1$ and $\hat{U}_2 < U_2$ hold. This means that $\hat{U}_1 - (U_2 - \delta_2(U_2 - U_1))$ is positive, therefore the value of $\hat{\delta}_1 \geq 0$ that minimizes the third term is $\hat{\delta}_1 = 0$, and that $\hat{U}_2 - U_2$ is negative, therefore the value of $\hat{\delta}_2 \geq 0$ that minimizes the second term is $\hat{\delta}_2 = 0$. This shows that the k that minimizes the cost (58) is a Dirac delta where $k[1] = \hat{\delta}_1 = 0$, $k[-1] = \hat{\delta}_2 = 0$ and $k[0] = 1$. \square

A.4 Proof of proposition (4.5)

Proof: Notice that If u^0 is a zero-mean signal, because total variation denoising preserves the mean of the original signal and $\sum_x k^0[x] = 1$ we have that also \hat{u} is a zero-mean signal. We can consider the different conditions on λ separately.

For a $\lambda \in [\lambda_{min}^l, \lambda_{max}^l]$ from Proposition 4.2 we have that the minimizer of problem (15) is

$$\hat{u}[x] = \begin{cases} \hat{U}_1 & x \in [-L_1, -2] \\ \hat{U}_2 & x \in [-1, L_2] \end{cases} \quad (59)$$

Since we can always express a zero-mean step as another scaled zero-mean step, we can write

$$\hat{u}[x] = au^0[x + 1] \quad (60)$$

for some constant a . We then solve the second step of the PAM algorithm

$$\hat{k} = \arg \min_k \|k * \hat{u} - f\|_2^2 \quad (61)$$

where we can write

$$\begin{aligned} &\|(k * \hat{u})[x] - f[x]\|_2^2 \\ &= \|a \sum_y k[y]u^0[x + 1 - y] - \sum_y k[y]u^0[x - y]\|_2^2 \\ &= \|a \sum_y k[y - 1]u^0[x - y] - \sum_y k^0[y]u^0[x - y]\|_2^2 \\ &= \|\sum_y (ak[y - 1] - k^0[y])u^0[x - y]\|_2^2 \end{aligned}$$

and have $\hat{k}[x - 1] = k^0[x]/a$. Finally, by applying the last two steps of the PAM algorithm one obtains $\hat{k}[x - 1] = k^0[x]$.

For $\lambda \in [\lambda_{min}^c, \lambda_{max}^c]$ from Proposition 4.2 we have that the minimizer of problem (15) is

$$\hat{u}[x] = \begin{cases} \hat{U}_1 & x \in [-L_1, -1] \\ \hat{U}_2 & x \in [0, L_2] \end{cases} \quad (62)$$

In this case we can write

$$\hat{u}[x] = au^0[x] \quad (63)$$

for some constant a . We then solve the second step of the PAM algorithm

$$\hat{k} = \arg \min_k \|k * \hat{u} - f\|_2^2 \quad (64)$$

where we can write

$$\begin{aligned} &\|(k * \hat{u})[x] - f[x]\|_2^2 \\ &= \|a \sum_y k[y]u^0[x - y] - \sum_y k[y]u^0[x - y]\|_2^2 \\ &= \|a \sum_y k[y]u^0[x - y] - \sum_y k^0[y]u^0[x - y]\|_2^2 \\ &= \|\sum_y (ak[y] - k^0[y])u^0[x - y]\|_2^2 \end{aligned} \quad (65)$$

and have $\hat{k}[x] = k^0[x]/a$. Finally, by applying the last two steps of the PAM algorithm one obtains $\hat{k}[x] = k^0[x]$.

Finally, for a $\lambda \in [\lambda_{min}^r, \lambda_{max}^r]$ from Proposition 4.2 we have that the minimizer of problem (15) is

$$\hat{u}[x] = \begin{cases} \hat{U}_1 & x \in [-L_1, 0] \\ \hat{U}_2 & x \in [1, L_2] \end{cases} \quad (66)$$

In this case we can write

$$\hat{u}[x] = au^0[x - 1] \quad (67)$$

for some constant a . We then solve the second step of the PAM algorithm

$$\hat{k} = \arg \min_k \|k * \hat{u} - f\|_2^2 \quad (68)$$

where we can write

$$\begin{aligned} &\|(k * \hat{u})[x] - f[x]\|_2^2 \\ &= \|a \sum_y k[y]u^0[x - 1 - y] - \sum_y k[y]u^0[x - y]\|_2^2 \\ &= \|a \sum_y k[y + 1]u^0[x - y] - \sum_y k^0[y]u^0[x - y]\|_2^2 \\ &= \|\sum_y (ak[y + 1] - k^0[y])u^0[x - y]\|_2^2 \end{aligned} \quad (69)$$

and have $\hat{k}[x + 1] = k^0[x]/a$. Finally, by applying the last two steps of the PAM algorithm one obtains $\hat{k}[x + 1] = k^0[x]$. \square

REFERENCES

- [1] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, 2006.
- [2] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *Signal Processing Magazine, IEEE*, vol. 13, no. 3, pp. 43–64, 1996.
- [3] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *ACM Trans. Graph.*, 2008, pp. 1–10.
- [4] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1–8, 2009.
- [5] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proceedings of the 11th European Conference on Computer Vision: Part I*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 157–170.
- [6] A. Levin, Y. Weiss, F. Durand, and W. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 2657–2664.
- [7] Y.-L. You and M. Kaveh, "A regularization approach to joint blur identification and image restoration," *Image Processing, IEEE Transactions on*, vol. 5, no. 3, pp. 416–428, 1996.
- [8] T. Chan and C.-K. Wong, "Total variation blind deconvolution," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370–375, 1998.

- [9] Y.-L. You and M. Kaveh, "Anisotropic blind image restoration," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 1, 1996, pp. 461–464 vol.2.
- [10] J. Huang and D. Mumford, "Statistics of natural images and models," pp. 541 – 547, 1999.
- [11] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding blind deconvolution algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2354–2367, 2011.
- [12] A. Tikhonov and V. Arsenin, *Solutions of ill-posed problems*. Vh Winston, 1977.
- [13] R. M. Rameshan, S. Chaudhuri, and R. Velmurugan, "Joint map estimation for blind deconvolution: When does it work?" in *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, ser. ICVGIP '12. New York, NY, USA: ACM, 2012, pp. 50:1–50:7.
- [14] T. C. Chan and C. Wong, "Convergence of the alternating minimization algorithm for blind deconvolution," *Linear Algebra and its Applications*, vol. 316, no. 13, pp. 259 – 285, 2000.
- [15] L. He, A. Marquina, and S. J. Osher, "Blind deconvolution using tv regularization and bregman iteration," *Int. J. Imaging Systems and Technology*, vol. 15, no. 1, pp. 74–83, 2005.
- [16] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physics D*, vol. 60, pp. 259–268, 1992.
- [17] C. Wang, L. Sun, Z. Chen, J. Zhang, and S. Yang, "Multi-scale blind motion deblurring using local minimum," *Inverse Problems*, vol. 26, no. 1, 2010.
- [18] D. P. Wipf and H. Zhang, "Analysis of bayesian blind deconvolution," in *EMMCVPR*, 2013, pp. 40–53.
- [19] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 1107–1114.
- [20] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 233–240.
- [21] J. Miskin and D. MacKay, "Ensemble learning for blind image separation and deconvolution," in *Advances in Independent Component Analysis*, ser. Perspectives in Neural Computing, M. Girolami, Ed. Springer London, 2000, pp. 123–141.
- [22] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos, "Bayesian blind deconvolution with general sparse image priors," in *ECCV*. Firenze, Italy: Springer, October 2012.
- [23] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, "A local/global approach to mesh parameterization."
- [24] D. Perrone and P. Favaro, "Total variation blind deconvolution: The devil is in the details," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [25] L. Condat, "A direct algorithm for l-d total variation denoising," *Signal Processing Letters, IEEE*, vol. 20, no. 11, pp. 1054–1057, 2013.
- [26] D. Strong and T. Chan, "Edge-preserving and scale-dependent properties of total variation regularization," *Inverse Problems*, vol. 19, no. 6, p. S165, 2003.
- [27] P. L. Davies and A. Kovac, "Local extremes, runs, strings and multiresolution," *The Annals of Statistics*, vol. 29, no. 1, pp. pp. 1–48, 2001.
- [28] T. Saito, Y. Takagaki, and T. Komatsu, "Three kinds of color total-variation semi-norms and its application to color-image denoising," in *ICIP*, 2011, pp. 1457–1460.
- [29] B. Goldlücke and D. Cremers, "An approach to vectorial total variation based on geometric measure theory," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 327–333.
- [30] P. Blomgren and T. Chan, *Color TV: Total Variation Methods for Restoration of Vector Valued Images*, ser. CAM report. Department of Mathematics, University of California, Los Angeles, 1996.
- [31] A. Levin, R. Fergus, F. Durand, and W. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, 2007.
- [32] L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *Computational Photography (ICCP), 2013 IEEE International Conference on*, April 2013, pp. 1–8.
- [33] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 479–486. [Online]. Available: <http://dx.doi.org/10.1109/ICCV.2011.6126278>
- [34] T. Michaeli and M. Irani, "Blind deblurring using internal patch recurrence," in *Computer Vision ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, vol. 8691, pp. 783–798.
- [35] D. Perrone, R. Diethelm, and P. Favaro, "Blind deconvolution via lower-bounded logarithmic image priors," in *Energy Minimization Methods in Computer Vision and Pattern Recognition - 10th International Conference, EMMCVPR 2015, Hong Kong, China, January 13-16, 2015. Proceedings*, 2014, pp. 112–125.
- [36] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 27–40.
- [37] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *CVPR*. IEEE, June 2009, pp. 1964–1971.
- [38] A. Goldstein and R. Fattal, "Blur-kernel estimation from spectral irregularities," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ser. ECCV'12, 2012, pp. 622–635.
- [39] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang, "Handling noise in single image deblurring using directional filters," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 612–619.
- [40] O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," in *Proceedings of the IEEE Workshop on Color and Photometry in Computer Vision, with ICCV 2011*, 2011.



Daniele Perrone received a M.Eng.degree in Artificial Intelligence from "Sapienza" Università di Roma in 2009 and a B.Eng degree from Università della Calabria in 2007. From June 2010 until June 2012 he has been a PhD candidate at Heriot Watt University. Since June 2012 he is a PhD candidate at University of Bern. His research interests are in computer vision, machine learning, image processing, inverse problems, variational methods and computational photography.



Paolo Favaro received the Laurea degree (BSc+MSc) from Università di Padova, Italy in 1999, and the M.Sc. and Ph.D. degree in electrical engineering from Washington University in St. Louis in 2002 and 2003 respectively. He was a postdoctoral researcher in the computer science department of the University of California, Los Angeles and subsequently in Cambridge University, UK. Between 2004 and 2006 he worked in medical imaging at Siemens Corporate Research, Princeton, USA. From 2006 to 2011 he was Lecturer and then Reader at Heriot-Watt University and Honorary Fellow at the University of Edinburgh, UK. In 2012 he became full professor at the University of Bern, Switzerland. His research interests are in computer vision, computational photography, machine learning, signal and image processing, estimation theory, inverse problems and variational techniques. He is a member of the IEEE.